A Probabilistic Approach to Syntax-based Reordering for Statistical Machine Translation

Chi-Ho Li, Dongdong Zhang, Mu Li, Ming Zhou
Microsoft Research Asia
Beijing, China
chl, dozhang@microsoft.com
muli, mingzhou@microsoft.com

Minghui Li, Yi Guan
Harbin Institute of Technology
Harbin, China
mhli@insun.hit.edu.cn
guanyi@insun.hit.edu.cn

Abstract

Inspired by previous preprocessing approaches to SMT, this paper proposes a novel, probabilistic approach to reordering which combines the merits of syntax and phrase-based SMT. Given a source sentence and its parse tree, our method generates, by tree operations, an n-best list of reordered inputs, which are then fed to standard phrase-based decoder to produce the optimal translation. Experiments show that, for the NIST MT-05 task of Chinese-to-English translation, the proposal leads to BLEU improvement of 1.56%.

1 Introduction

The phrase-based approach has been considered the default strategy to Statistical Machine Translation (SMT) in recent years. It is widely known that the phrase-based approach is powerful in local lexical choice and word reordering within short distance. However, long-distance reordering is problematic in phrase-based SMT. For example, the distance-based reordering model (Koehn et al., 2003) allows a decoder to translate in non-monotonous order, under the constraint that the distance between two phrases translated consecutively does not exceed a limit known as *distortion limit*. In theory the distortion limit can be assigned a very large value so that all possible reorderings are allowed, yet in practise it is observed that too high a distortion limit not only harms efficiency but also translation performance (Koehn et al., 2005). In our own experiment setting, the best distortion limit for Chinese-English translation is 4. However, some ideal translations exhibit reorderings longer than such distortion limit. Consider the sentence pair in NIST MT-2005 test set shown in figure 1(a): after translating the word "修补/*mend*", the decoder should 'jump' across six words and translate the last phrase "关系 裂缝/*fissures in the relationship*". Therefore, while short-distance reordering is under the scope of the distance-based model, long-distance reordering is simply out of the question.

A terminological remark: In the rest of the paper, we will use the terms *global reordering* and *local reordering* in place of long-distance reordering and short-distance reordering respectively. The distinction between long and short distance reordering is solely defined by distortion limit.

Syntax[1] is certainly a potential solution to global reordering. For example, for the last two Chinese phrases in figure 1(a), simply swapping the two children of the NP node will produce the correct word order on the English side. However, there are also reorderings which do not agree with syntactic analysis. Figure 1(b) shows how our phrase-based decoder[2] obtains a good English translation by reordering two blocks. It should be noted that the second Chinese block "结束 时" and its English counterpart "*at the end of*" are not constituents at all.

In this paper, our interest is the value of syntax in reordering, and the major statement is that syntactic information is useful in handling global reordering

[1]Here by syntax it is meant linguistic syntax rather than formal syntax.

[2]The decoder is introduced in section 6.

Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pages 720–727,
Prague, Czech Republic, June 2007.

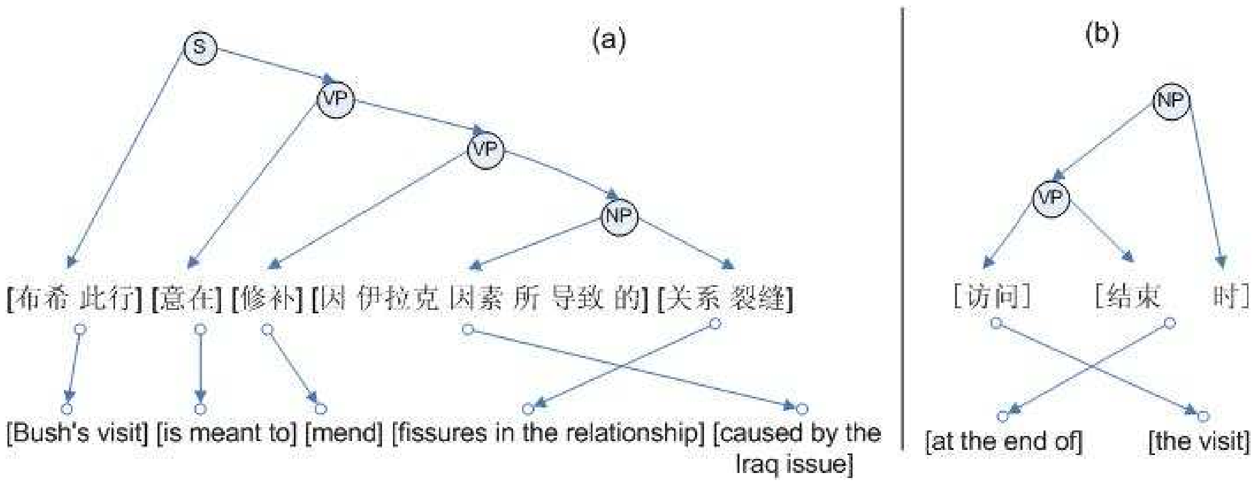

Figure 1: Examples on how syntax (a) helps and (b) harms reordering in Chinese-to-English translation
The lines and nodes on the top half of the figures show the phrase structure of the Chinese sentences, while the links on the bottom half of the figures show the alignments between Chinese and English phrases. Square brackets indicate the boundaries of blocks found by our decoder.

and it achieves better MT performance on the basis of the standard phrase-based model. To prove it, we developed a hybrid approach which preserves the strength of phrase-based SMT in local reordering as well as the strength of syntax in global reordering.

Our method is inspired by previous *preprocessing* approaches like (Xia and McCord, 2004), (Collins et al., 2005), and (Costa-jussà and Fonollosa, 2006), which split translation into two stages:

$$S \rightarrow S' \rightarrow T \tag{1}$$

where a sentence of the source language (SL), S, is first reordered with respect to the word order of the target language (TL), and then the reordered SL sentence S' is translated as a TL sentence T by monotonous translation.

Our first contribution is a new translation model as represented by formula 2:

$$S \rightarrow n \times S' \rightarrow n \times T \rightarrow \hat{T} \tag{2}$$

where an n-best list of S', instead of only one S', is generated. The reason of such change will be given in section 2. Note also that the translation process $S' \rightarrow T$ is not monotonous, since the distance-based model is needed for local reordering. Our second contribution is our definition of the best translation:

$$\arg\max_T \exp(\lambda_r \log Pr(S \rightarrow S') + \sum_i \lambda_i F_i(S' \rightarrow T))$$

where F_i are the features in the standard phrase-based model and $Pr(S \rightarrow S')$ is our new feature, viz. the probability of reordering S as S'. The details of this model are elaborated in sections 3 to 6. The settings and results of experiments on this new model are given in section 7.

2 Related Work

There have been various attempts to syntax-based SMT, such as (Yamada and Knight, 2001) and (Quirk et al., 2005). We do not adopt these models since a lot of subtle issues would then be introduced due to the complexity of syntax-based decoder, and the impact of syntax on reordering will be difficult to single out.

There have been many reordering strategies under the phrase-based camp. A notable approach is lexicalized reordering (Koehn et al., 2005) and (Tillmann, 2004). It should be noted that this approach achieves the best result within certain distortion limit and is therefore not a good model for global reordering.

There are a few attempts to the preprocessing approach to reordering. The most notable ones are (Xia and McCord, 2004) and (Collins et al., 2005), both of which make use of linguistic syntax in the preprocessing stage. (Collins et al., 2005) analyze German clause structure and propose six types of rules for transforming German parse trees with respect to English word order. Instead of relying on manual rules, (Xia and McCord, 2004) propose a method in learning patterns of rewriting SL sentences. This method parses training data and uses some heuristics to align SL phrases with TL ones. From such alignment it can extract rewriting patterns, of which the units are words and POSs. The learned rewriting rules are then applied to rewrite SL sentences before monotonous translation.

Despite the encouraging results reported in these papers, the two attempts share the same shortcoming that their reordering is deterministic. As pointed out in (Al-Onaizan and Papineni, 2006), these strategies make hard decisions in reordering which cannot be undone during decoding. That is, the choice of reordering is independent from other translation factors, and once a reordering mistake is made, it cannot be corrected by the subsequent decoding.

To overcome this weakness, we suggest a method to 'soften' the hard decisions in preprocessing. The essence is that our preprocessing module generates n-best S's rather than merely one S'. A variety of reordered SL sentences are fed to the decoder so that the decoder can consider, to certain extent, the interaction between reordering and other factors of translation. The entire process can be depicted by formula 2, recapitulated as follows:

$$S \rightarrow n \times S' \rightarrow n \times T \rightarrow \hat{T}.$$

Apart from their deterministic nature, the two previous preprocessing approaches have their own weaknesses. (Collins et al., 2005) count on manual rules and it is suspicious if reordering rules for other language pairs can be easily made. (Xia and McCord, 2004) propose a way to learn rewriting patterns, nevertheless the units of such patterns are words and their POSs. Although there is no limit to the length of rewriting patterns, due to data sparseness most patterns being applied would be short ones. Many instances of global reordering are therefore left unhandled.

3 The Acquisition of Reordering Knowledge

To avoid this problem, we give up using rewriting patterns and design a form of reordering knowledge which can be directly applied to parse tree nodes. Given a node N on the parse tree of an SL sentence, the required reordering knowledge should enable the preprocessing module to determine how probable the children of N are reordered.[3] For simplicity, let us first consider the case of binary nodes only. Let N_1 and N_2, which yield phrases p_1 and p_2 respectively, be the child nodes of N. We want to determine the order of p_1 and p_2 with respect to their TL counterparts, $T(p_1)$ and $T(p_2)$. The knowledge for making such a decision can be learned from a word-aligned parallel corpus. There are two questions involved in obtaining training instances:

- How to define $T(p_i)$?
- How to define the order of $T(p_i)$s?

For the first question, we adopt a similar method as in (Fox, 2002): given an SL phrase $p_s = s_1 \ldots s_i \ldots s_n$ and a word alignment matrix A, we can enumerate the set of TL words $\{t_i : t_i \epsilon A(s_i)\}$, and then arrange the words in the order as they appear in the TL sentence. Let $\mathit{first}(t)$ be the first word in this sorted set and $\mathit{last}(t)$ be the last word. $T(p_s)$ is defined as the phrase $\mathit{first}(t) \ldots \mathit{last}(t)$ in the TL sentence. Note that $T(p_s)$ may contain words not in the set $\{t_i\}$.

The question of the order of two TL phrases is not a trivial one. Since a word alignment matrix usually contains a lot of noises as well as one-to-many and many-to-many alignments, two TL phrases may overlap with each other. For the sake of the quality of reordering knowledge, if $T(p_1)$ and $T(p_2)$ overlap, then the node N with children N_1 and N_2 is not taken as a training instance. Obviously it will greatly reduce the amount of training input. To remedy data sparseness, less probable alignment points are removed so as to minimize overlapping phrases, since, after removing some alignment point, one of the TL phrases may become shorter and the two phrases may no longer overlap. The implementation is similar to the idea of *lexical weight* in (Koehn et al., 2003): all points in the alignment matrices of the entire training corpus are collected to calculate the probabilistic distribution, $P(t|s)$, of some TL word

[3] Some readers may prefer the expression *the subtree rooted at node N* to *node N*. The latter term is used in this paper for simplicity.

t given some SL word s. Any pair of overlapping $T(p_i)$s will be redefined by iteratively removing less probable word alignments until they no longer overlap. If they still overlap after all one/many-to-many alignments have been removed, then the refinement will stop and N, which covers p_is, is no longer taken as a training instance.

In sum, given a bilingual training corpus, a parser for the SL, and a word alignment tool, we can collect all binary parse tree nodes, each of which may be an instance of the required reordering knowledge. The next question is what kind of reordering knowledge can be formed out of these training instances. Two forms of reordering knowledge are investigated:

1. Reordering Rules, which have the form

$$Z:\ X\,Y \Rightarrow \begin{cases} X\,Y & Pr(\text{IN-ORDER}) \\ Y\,X & Pr(\text{INVERTED}) \end{cases}$$

   where Z is the phrase label of a binary node and X and Y are the phrase labels of Z's children, and $Pr(\text{INVERTED})$ and $Pr(\text{IN-ORDER})$ are the probability that X and Y are inverted on TL side and that not inverted, respectively. The probability figures are estimated by Maximum Likelihood Estimation.

2. Maximum Entropy (ME) Model, which does the binary classification whether a binary node's children are inverted or not, based on a set of features over the SL phrases corresponding to the two children nodes. The features that we investigated include the leftmost, rightmost, head, and context words[4], and their POSs, of the SL phrases, as well as the phrase labels of the SL phrases and their parent.

4 The Application of Reordering Knowledge

After learning reordering knowledge, the preprocessing module can apply it to the parse tree, t_S, of an SL sentence S and obtain the n-best list of S'. Since a ranking of S' is needed, we need some way to score each S'. Here probability is used as the scoring metric. In this section it is explained how the n-best reorderings of S and their associated scores/probabilites are computed.

Let us first look into the scoring of a particular reordering. Let $Pr(p \rightarrow p')$ be the probability of reordering a phrase p into p'. For a phrase q yielded by a non-binary node, there is only one 'reordering' of q, viz. q itself, thus $Pr(q \rightarrow q) = 1$. For a phrase p yielded by a binary node N, whose left child N_1 has reorderings p_1^i and right child N_2 has the reorderings p_2^j ($1 \leq i, j \leq n$), p' has the form $p_1^i p_2^j$ or $p_2^j p_1^i$. Therefore, $Pr(p \rightarrow p') =$

$$\begin{cases} Pr(\text{IN-ORDER}) \times Pr(p_1^i \rightarrow p_1^{i'}) \times Pr(p_2^j \rightarrow p_2^{j'}) \\ Pr(\text{INVERTED}) \times Pr(p_2^j \rightarrow p_2^{j'}) \times Pr(p_1^i \rightarrow p_1^{i'}) \end{cases}$$

The figures $Pr(\text{IN-ORDER})$ and $Pr(\text{INVERTED})$ are obtained from the learned reordering knowledge. If reordering knowledge is represented as rules, then the required probability is the probability associated with the rule that can apply to N. If reordering knowledge is represented as an ME model, then the required probability is:

$$P(r|N) = \frac{\exp(\sum_i \lambda_i f_i(N, r))}{\sum_{r'} \exp(\sum_i \lambda_i f_i(N, r'))}$$

where $r \epsilon \{\text{IN-ORDER}, \text{INVERTED}\}$, and f_i's are features used in the ME model.

Let us turn to the computation of the n-best reordering list. Let $R(N)$ be the number of reorderings of the phrase yielded by N, then:

$$R(N) = \begin{cases} 2R(N_1)R(N_2) & \text{if } N \text{ has children } N_1, N_2 \\ 1 & \text{otherwise} \end{cases}$$

It is easily seen that the number of S's increases exponentially. Fortunately, what we need is merely an n-best list rather than a full list of reorderings. Starting from the leaves of t_S, for each node N covering phrase p, we only keep track of the n p's that have the highest reordering probability. Thus $R(N) \leq n$. There are at most $2n^2$ reorderings for any node and only the top-scored n reorderings are recorded. The n-best reorderings of S, i.e. the n-best reorderings of the yield of the root node of t_S, can be obtained by this efficient bottom-up method.

5 The Generalization of Reordering Knowledge

In the last two sections reordering knowledge is learned from and applied to binary parse tree nodes

[4] The context words of the SL phrases are the word to the left of the left phrase and the word to the right of the right phrase.

only. It is not difficult to generalize the theory of reordering knowledge to nodes of other branching factors. The case of binary nodes is simple as there are only two possible reorderings. The case of 3-ary nodes is a bit more complicated as there are six.[5] In general, an n-ary node has $n!$ possible reorderings of its children. The maximum entropy model has the same form as in the binary case, except that there are more classes of reordering patterns as n increases. The form of reordering rules, and the calculation of reordering probability for a particular node, can also be generalized easily.[6] The only problem for the generalized reordering knowledge is that, as there are more classes, data sparseness becomes more severe.

6 The Decoder

The last three sections explain how the $S \rightarrow n \times S'$ part of formula 2 is done. The $S' \rightarrow T$ part is simply done by our re-implementation of PHARAOH (Koehn, 2004). Note that non-monotonous translation is used here since the distance-based model is needed for local reordering. For the $n \times T \rightarrow \hat{T}$ part, the factors in consideration include the score of T returned by the decoder, and the reordering probability $Pr(S \rightarrow S')$. In order to conform to the log-linear model used in the decoder, we integrate the two factors by defining the total score of T as formula 3:

$$\exp(\lambda_r \log Pr(S \rightarrow S') + \sum_i \lambda_i F_i(S' \rightarrow T)) \quad (3)$$

The first term corresponds to the contribution of syntax-based reordering, while the second term that of the features F_i used in the decoder. All the feature weights (λs) were trained using our implementation of Minimum Error Rate Training (Och, 2003). The final translation $\hat{T}$ is the T with the highest total score.

[5] Namely, $N_1N_2N_3$, $N_1N_3N_2$, $N_2N_1N_3$, $N_2N_3N_1$, $N_3N_1N_2$, and $N_3N_2N_1$, if the child nodes in the original order are N_1, N_2, and N_3.

[6] For example, the reordering probability of a phrase $p = p_1p_2p_3$ generated by a 3-ary node N is

$$Pr(r) \times Pr(p_1^i) \times Pr(p_2^j) \times Pr(p_3^k)$$

where r is one of the six reordering patterns for 3-ary nodes.

It is observed in pilot experiments that, for a lot of long sentences containing several clauses, only one of the clauses is reordered. That is, our greedy reordering algorithm (c.f. section 4) has a tendency to focus only on a particular clause of a long sentence.

The problem was remedied by modifying our decoder such that it no longer translates a sentence at once; instead the new decoder does:

1. split an input sentence S into clauses $\{C_i\}$;
2. obtain the reorderings among $\{C_i\}$, $\{S_j\}$;
3. for each S_j, do
   (a) for each clause C_i in S_j, do
      i. reorder C_i into n-best C_i's,
      ii. translate each C_i' into $T(C_i')$,
      iii. select $\hat{T}(C_i')$;
   (b) concatenate $\{\hat{T}(C_i')\}$ into T_j;
4. select $\hat{T}_j$.

Step 1 is done by checking the parse tree if there are any IP or CP nodes[7] immediately under the root node. If yes, then all these IPs, CPs, and the remaining segments are treated as clauses. If no, then the entire input is treated as one single clause. Step 2 and step 3(a)(i) still follow the algorithm in section 4. Step 3(a)(ii) is trivial, but there is a subtle point about the calculation of language model score: the language model score of a translated clause is not independent from other clauses; it should take into account the last few words of the previous translated clause. The best translated clause $\hat{T}(C_i')$ is selected in step 3(a)(iii) by equation 3. In step 4 the best translation $\hat{T}_j$ is

$$\arg\max_{T_j} \exp(\lambda_r \log Pr(S \rightarrow S_j) + \sum_i score(T(C_i'))).$$

7 Experiments

7.1 Corpora

Our experiments are about Chinese-to-English translation. The NIST MT-2005 test data set is used for evaluation. (Case-sensitive) BLEU-4 (Papineni et al., 2002) is used as the evaluation metric. The

[7] IP stands for *inflectional phrase* and CP for *complementizer phrase*. These two types of phrases are *clauses* in terms of the Government and Binding Theory.

Branching Factor	2	3	>3
Count	12294	3173	1280
Percentage	73.41	18.95	7.64

Table 1: Distribution of Parse Tree Nodes with Different Branching Factors Note that nodes with only one child are excluded from the survey as reordering does not apply to such nodes.

test set and development set of NIST MT-2002 are merged to form our development set. The training data for both reordering knowledge and translation table is the one for NIST MT-2005. The GIGAWORD corpus is used for training language model. The Chinese side of all corpora are segmented into words by our implementation of (Gao et al., 2003).

7.2 The Preprocessing Module

As mentioned in section 3, the preprocessing module for reordering needs a parser of the SL, a word alignment tool, and a Maximum Entropy training tool. We use the Stanford parser (Klein and Manning, 2003) with its default Chinese grammar, the GIZA++ (Och and Ney, 2000) alignment package with its default settings, and the ME tool developed by (Zhang, 2004).

Section 5 mentions that our reordering model can apply to nodes of any branching factor. It is interesting to know how many branching factors should be included. The distribution of parse tree nodes as shown in table 1 is based on the result of parsing the Chinese side of NIST MT-2002 test set by the Stanford parser. It is easily seen that the majority of parse tree nodes are binary ones. Nodes with more than 3 children seem to be negligible. The 3-ary nodes occupy a certain proportion of the distribution, and their impact on translation performance will be shown in our experiments.

7.3 The decoder

The data needed by our Pharaoh-like decoder are translation table and language model. Our 5-gram language model is trained by the SRI language modeling toolkit (Stolcke, 2002). The translation table is obtained as described in (Koehn et al., 2003), i.e. the alignment tool GIZA++ is run over the training data in both translation directions, and the two alignment matrices are integrated by the GROW-DIAG-FINAL method into one matrix, from which phrase translation probabilities and lexical weights of both directions are obtained.

Test	Setting	BLEU
B1	standard phrase-based SMT	29.22
B2	(B1) + clause splitting	29.13

Table 2: Experiment Baseline

Test	Setting	BLEU 2-ary	BLEU 2,3-ary
1	rule	29.77	30.31
2	ME (phrase label)	29.93	30.49
3	ME (left,right)	30.10	30.53
4	ME ((3)+head)	30.24	30.71
5	ME ((3)+phrase label)	30.12	30.30
6	ME ((4)+context)	30.24	30.76

Table 3: Tests on Various Reordering Models
The 3rd column comprises the BLEU scores obtained by reordering binary nodes only, the 4th column the scores by reordering both binary and 3-ary nodes. The features used in the ME models are explained in section 3.

The most important system parameter is, of course, distortion limit. Pilot experiments using the standard phrase-based model show that the optimal distortion limit is 4, which was therefore selected for all our experiments.

7.4 Experiment Results and Analysis

The baseline of our experiments is the standard phrase-based model, which achieves, as shown by table 2, the BLEU score of 29.22. From the same table we can also see that the clause splitting mechanism introduced in section 6 does not significantly affect translation performance.

Two sets of experiments were run. The first set, of which the results are shown in table 3, tests the effect of different forms of reordering knowledge. In all these tests only the top 10 reorderings of each clause are generated. The contrast between tests 1 and 2 shows that ME modeling of reordering outperforms reordering rules. Tests 3 and 4 show that phrase labels can achieve as good performance as the lexical features of mere leftmost and rightmost words. However, when more lexical features

Input	海南省 2005年 还 将 继续 增加 对 公共 服务 和 社会 事业 基础 设施 投资
Reference	Hainan province will continue to increase its investment in the public services and social services infrastructures in 2005
Baseline	Hainan Province in 2005 will continue to increase for the public service and social infrastructure investment
Translation with Preprocessing	Hainan Province in 2005 will continue to increase investment in public services and social infrastructure

Table 4: Translation Example 1

Test	Setting	BLEU
a	length constraint	30.52
b	DL=0	30.48
c	n=100	30.78

Table 5: Tests on Various Constraints

are added (tests 4 and 6), phrase labels can no longer compete with lexical features. Surprisingly, test 5 shows that the combination of phrase labels and lexical features is even worse than using either phrase labels or lexical features only.

Apart from quantitative evaluation, let us consider the translation example of test 6 shown in table 4. To generate the correct translation, a phrase-based decoder should, after translating the word "增加" as "*increase*", jump to the last word "投资(investment)". This is obviously out of the capability of the baseline model, and our approach can accomplish the desired reordering as expected.

By and large, the experiment results show that no matter what kind of reordering knowledge is used, the preprocessing of syntax-based reordering does greatly improve translation performance, and that the reordering of 3-ary nodes is crucial.

The second set of experiments test the effect of some constraints. The basic setting is the same as that of test 6 in the first experiment set, and reordering is applied to both binary and 3-ary nodes. The results are shown in table 5.

In test (a), the constraint is that the module does not consider any reordering of a node if the yield of this node contains not more than four words. The underlying rationale is that reordering within distortion limit should be left to the distance-based model during decoding, and syntax-based reordering should focus on global reordering only. The result shows that this hypothesis does not hold. In practice syntax-based reordering also helps local reordering. Consider the translation example of test (a) shown in table 6. Both the baseline model and our model translate in the same way up to the word "围绕" (which is incorrectly translated as "*and*"). From this point, the proposed preprocessing model correctly jump to the last phrase "进行 了 讨论/*discussed*", while the baseline model fail to do so for the best translation. It should be noted, however, that there are only four words between "围绕" and the last phrase, and the desired order of decoding is within the capability of the baseline system. With the feature of syntax-based global reordering, a phrase-based decoder performs better even with respect to local reordering. It is because syntax-based reordering adds more weight to a hypothesis that moves words across longer distance, which is penalized by the distance-based model.

In test (b) distortion limit is set as 0; i.e. reordering is done merely by syntax-based preprocessing. The worse result is not surprising since, after all, preprocessing discards many possibilities and thus reduce the search space of the decoder. Some local reordering model is still needed during decoding.

Finally, test (c) shows that translation performance does not improve significantly by raising the number of reorderings. This implies that our approach is very efficient in that only a small value of n is capable of capturing the most important global reordering patterns.

8 Conclusion and Future Work

This paper proposes a novel, probabilistic approach to reordering which combines the merits of syntax and phrase-based SMT. On the one hand, global reordering, which cannot be accomplished by the

Input	与此同时 , 尤先科 和 助手 围绕 组建 新 政府 问题 进行 了 讨论
Reference	Meanwhile , Yushchenko and his assistants discussed issues concerning the establishment of a new government
Baseline	The same time , Yushchenko assistants and a new Government on issues discussed
Translation with Preprocessing	The same time , Yushchenko assistants and held discussions on the issue of a new government

Table 6: Translation Example 2

phrase-based model, is enabled by the tree operations in preprocessing. On the other hand, local reordering is preserved and even strengthened in our approach. Experiments show that, for the NIST MT-05 task of Chinese-to-English translation, the proposal leads to BLEU improvement of 1.56%.

Despite the encouraging experiment results, it is still not very clear how the syntax-based and distance-based models complement each other in improving word reordering. In future we need to investigate their interaction and identify the contribution of each component. Moreover, it is observed that the parse trees returned by a full parser like the Stanford parser contain too many nodes which seem not be involved in desired reorderings. Shallow parsers should be tried to see if they improve the quality of reordering knowledge.

References

Yaser Al-Onaizan, and Kishore Papineni. 2006. Distortion Models for Statistical Machine Translation. *Proceedings for ACL 2006*.

Michael Collins, Philipp Koehn, and Ivona Kucerova. 2005. Clause Restructuring for Statistical Machine Translation. *Proceedings for ACL 2005*.

M.R. Costa-jussà, and J.A.R. Fonollosa. 2006. Statistical Machine Reordering. *Proceedings for EMNLP 2006*.

Heidi Fox. 2002. Phrase Cohesion and Statistical Machine Translation. *Proceedings for EMNLP 2002*.

Jianfeng Gao, Mu Li, and Chang-Ning Huang 2003. Improved Source-Channel Models for Chinese Word Segmentation. *Proceedings for ACL 2003*.

Dan Klein and Christopher D. Manning. 2003. Accurate Unlexicalized Parsing. *Proceedings for ACL 2003*.

Philipp Koehn, Franz J. Och, and Daniel Marcu. 2003. Statistical Phrase-based Translation. *Proceedings for HLT-NAACL 2003*.

Philipp Koehn. 2004. Pharaoh: a Beam Search Decoder for Phrase-Based Statistical Machine Translation Models. *Proceedings for AMTA 2004*.

Philipp Koehn, Amittai Axelrod, Alexandra Birch Mayne, Chris Callison-Burch, Miles Osborne, and David Talbot 2005. Edinburgh System Description for the 2005 IWSLT Speech Translation Evaluation. *Proceedings for IWSLT 2005*.

Franz J. Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. *Proceedings for ACL 2003*.

Franz J. Och, and Hermann Ney. 2000. Improved Statistical Alignment Models. *Proceedings for ACL 2000*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. *Proceedings for ACL 2002*.

Chris Quirk, Arul Menezes, and Colin Cherry. 2005. Dependency Treelet Translation: Syntactically Informed Phrasal SMT. *Proceedings for ACL 2005*.

Andreas Stolcke. 2002. SRILM - An Extensible Language Modeling Toolkit. *Proceedings for the International Conference on Spoken Language Understanding 2002*.

Christoph Tillmann. 2004. A Unigram Orientation Model for Statistical Machine Translation. *Proceedings for ACL 2004*.

Fei Xia, and Michael McCord 2004. Improving a Statistical MT System with Automatically Learned Rewrite Patterns. *Proceedings for COLING 2004*.

Kenji Yamada, and Kevin Knight. 2001. A syntax-based statistical translation model. *Proceedings for ACL 2001*.

Le Zhang. 2004. Maximum Entropy Modeling Toolkit for Python and C++. http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html.